*Florida International University*

*School of Computing and Information Sciences*

Software Engineering Focus

Feature Document

User Story #**134**

**Team Member:**

Nelson Cruz

**Product Owner(s)**:

Mohsen Taheri

**Mentor(s)**:

Mohsen Taheri

**Instructor**: Masoud Sadjadi

**User Story** Implement User Password Reset

* As a User I would like to reset my password so that I can login to the Neat system

**Acceptance Criteria**

* There exists a mechanism at the front end to allow a user the request for a new password
* After a user’s request, a password reset code is sent to the user’s registered email address
* There exists a mechanism at the front end for user to enter a new password is created
* At the backend new password is stored
* The user is taken to the login page

**Use Case #001-Update user’s password**

**Actors**

Initiated by Neat User

**Entry condition**

The user is at the login

**Flow of events**

1. While at the login screen, user clicks on the **forgot** link

2. The user is prompted with the forgot password screen

3. User enters his/her email address, then click request and is taken to the reset password page

4. An email is sent to user’s primary email address with a validation code

5. User enters code, password, and email at the reset password page

6. After user clicks on reset, he/she is taken to the login page

**Alternate flow of events**

* 3a.
* The user enters an incorrect email address and is prompted with a warning message
* 5a.
* User enters invalid information and is prompted with a warning message

**Exit condition**

The user enters a valid code and is taken to the Login page

**Quality requirements**

The user is acknowledges of success or failure to register within 5 seconds.

C:\Users\ncruz\Downloads\Password_reset_usecase.png

**Use Case Diagram**

OS X:Users:macbook:Downloads:neat-pasword-reset.png

**Sequence Diagram**

OS X:Users:macbook:Pictures:reset_password_class.png

**Class Diagram**

**Unit Test**

**Test Case 1 (Sunny Day)**

**Purpose**

* Ensure the system validates a current user

**Precondition**

* A valid email address is entered by user, the user exists in the system

**Input**

* User’s primary email address

**Expected Result**

* An e-mail is sent to the user’s primary e-mail address

**Actual Result**

* An e-mail Is received by user with validation code

**Test Case 2 (Rainy Day)**

**Purpose**

* Ensure the system rejects an invalid user

**Precondition**

* An invalid email address is entered by user, the user does not exists in the system

**Input**

* invalid email address

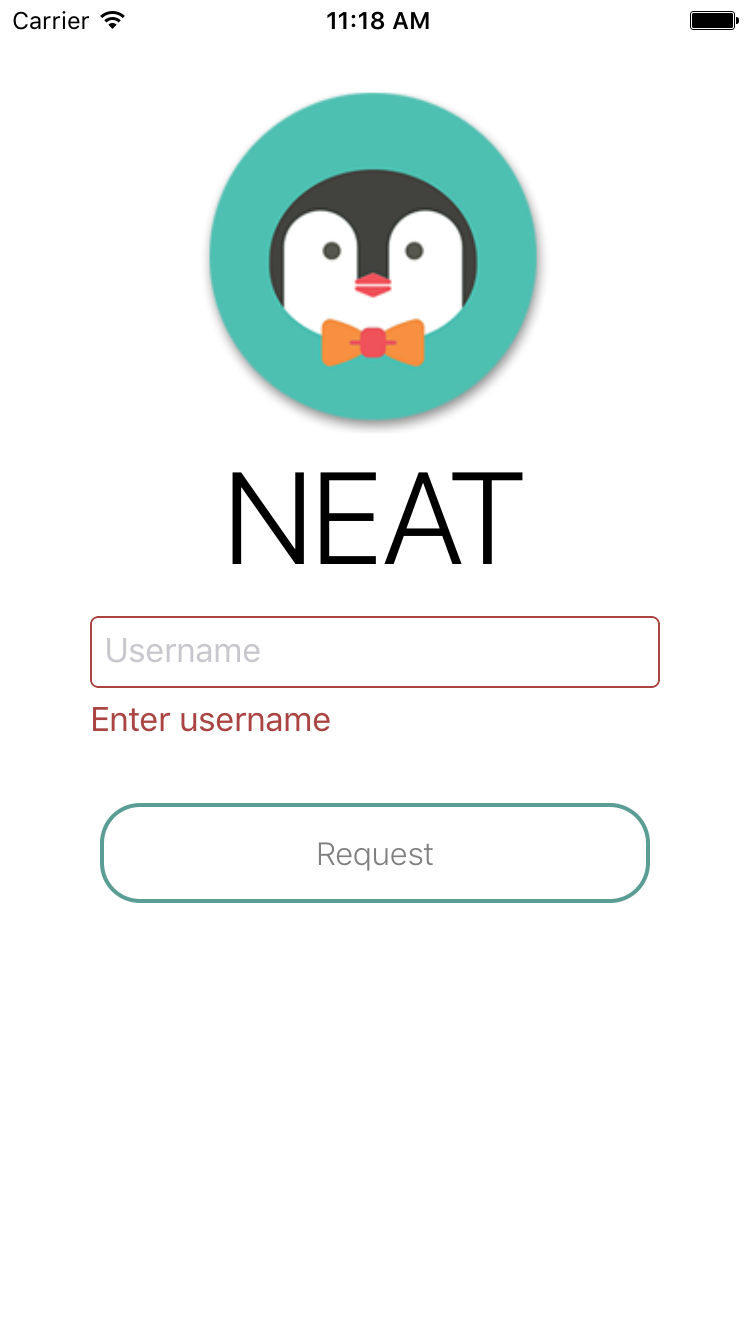
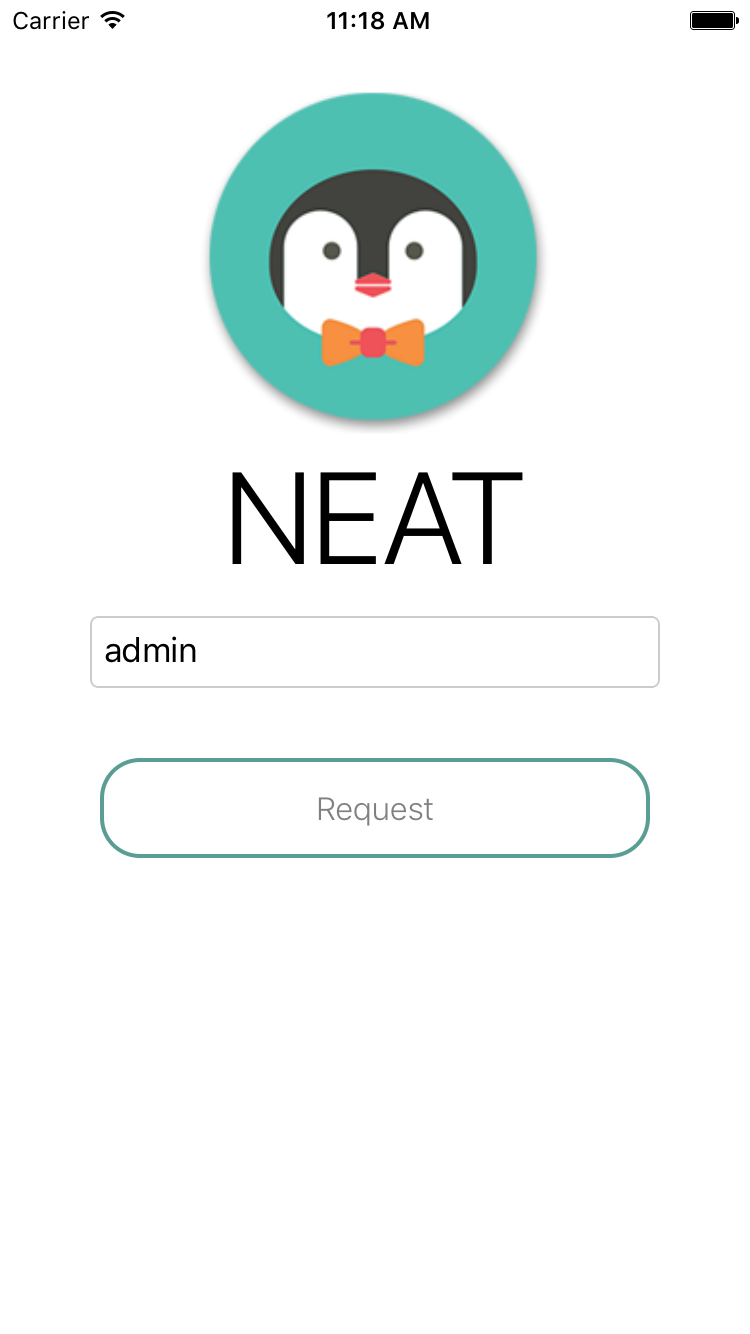
**Expected Result**

* Prompt the user to enter a valid email message

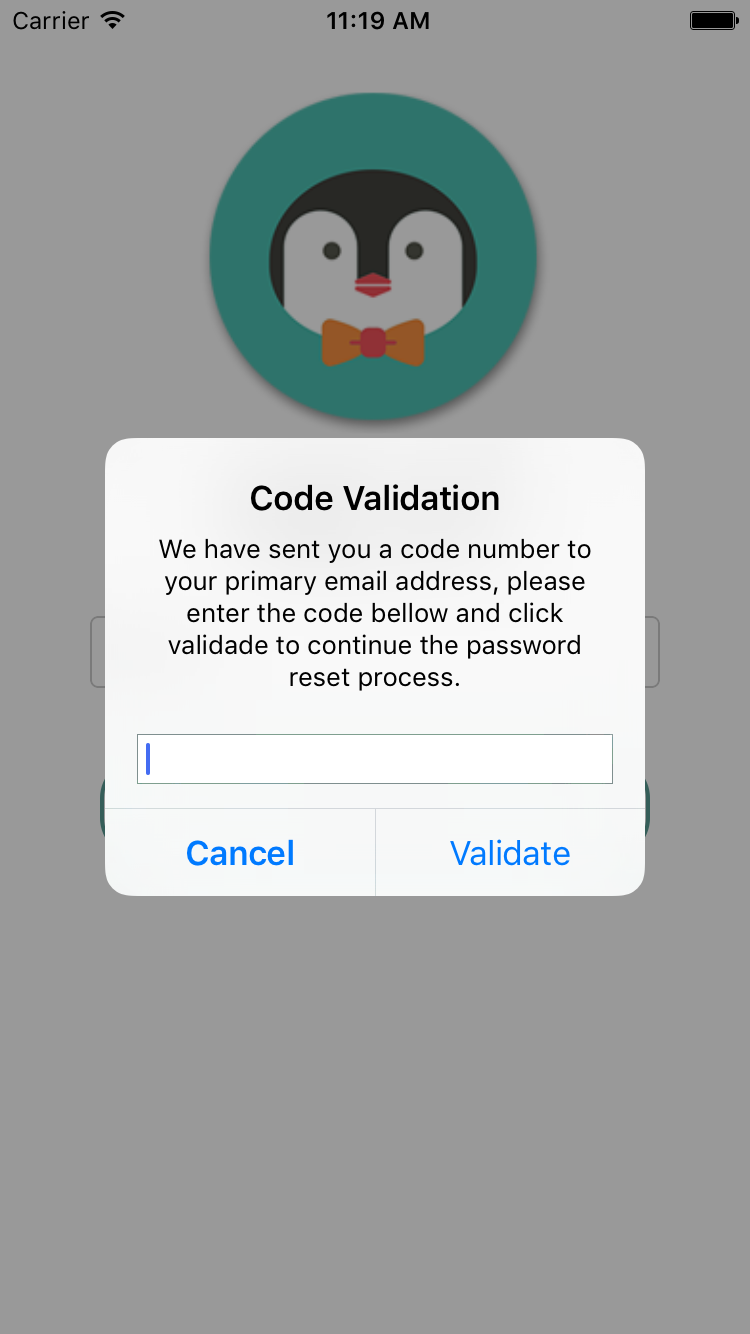
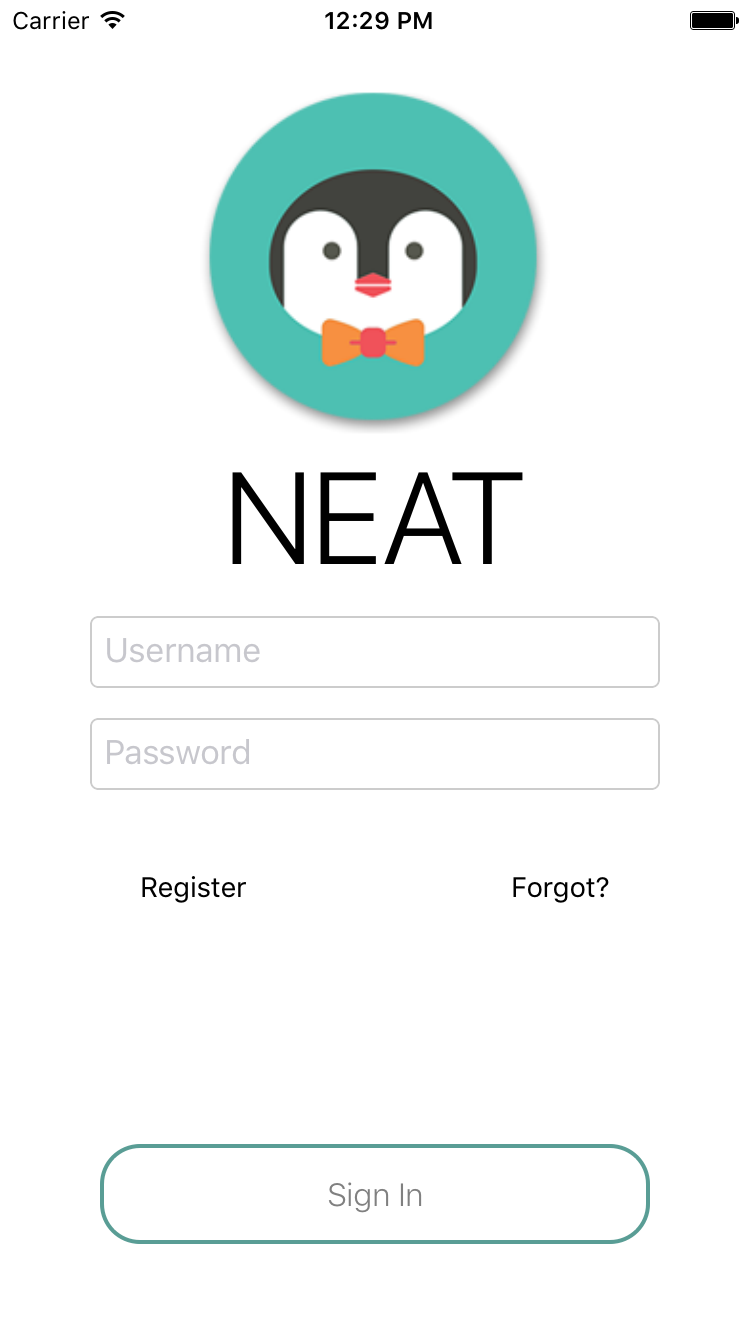
**Actual Result**

* no e-mail Is received by user

**Visual User Guide**

Request password invalid information Request password valid information

Enter validation code sent to email Go to dashboard after valid code entered